CSE211Dijital Tasarım

Akdeniz Üniversitesi

Hafta04-05: Boole Cebiri ve Mantık Kapıları

Doç.Dr. Taner Danışman

tdanisman@akdeniz.edu.tr

Ders programı

Hafta 01	09/16/2024	Giriş
Hafta 02	23/09/2024	Dijital Sistemler ve İkili Sayılar I
Hafta 03	30/09/2024	Dijital Sistemler ve İkili Sayılar II
Hafta 04	10/07/2024	Boole Cebiri ve Mantık Kapıları I
Hafta 05	10/14/2024	Boole Cebiri ve Mantık Kapıları II
Hafta 06	10/21/2024	Kapı Seviyesi Minimizasyonu
Hafta 07	10/28/2024	Karnaugh Haritaları
Hafta 08	11/04/2024	Vize
Hafta 09	11/11/2024	Karnaugh Haritaları
Hafta 10	11/18/2024	Kombinasyonel Mantık
11. Hafta	25.11.2024	Kombinasyonel Mantık
12. Hafta	12/02/2024	Zamanlama, gecikmeler ve tehlikeler
Hafta 13	12/09/2024	Eşzamanlı Sıralı Mantık
Hafta 14	12/16/2024	Eşzamanlı Sıralı Mantık

Önceki bölümde bilinmesi gereken terimler

DVD (Dijital Çok Yönlü Disk)

GUI (Grafiksel Kullanıcı Arayüzü) BCD

(İkili Kodlanmış Ondalık) ASCII (Bilgi

Değişimi için Amerikan Standart Kodu) STX (Metnin Başlangıcı) ETX (Metnin

Sonu) CR (Satır Başı)

HDL (Donanım

Açıklama Dili)

Boole Cebiri

Boole cebiri, iki değerden birine sahip olabilen değişkenlerin işlenmesine yarayan bir matematiksel sistemdir .

Biçimsel mantıkta bu değerler "doğru"dur ve "YANLIŞ."

Dijital sistemlerde bu değerler "açık" ve "kapalı"dır, 1 ve 0, ya da “yüksek” ve “düşük”.

Boolean ifadeleri,
Boolean değişkenleri üzerinde işlemler.

Yaygın Boole operatörleri arasında AND, OR bulunur.
ve DEĞİL.

Boole Cebiri – Doğruluk Tablosu

Bir Boole operatörü, bir doğruluk tablosu kullanılarak tam olarak tanımlanabilir.

Boolean için doğruluk tablosu

Sağ tarafta AND ve OR operatörleri gösterilmektedir.

AND operatörü aynı zamanda bir Boole ürünü olarak da bilinir. OR operatörü Boole toplamıdır.

X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

Boole Cebiri – Doğruluk Tablosu

Gerçek tablosu

Boolean NOT operatörü sağda gösterilmektedir.

NOT işlemi çoğunlukla bir üst çizgi ile belirtilir . Bazen bir asal işaret (') veya bir "dirsek" ( ) ile belirtilir .

NOT x

x	$\overline{x}$
0	1
1	0

Boole Cebiri – Boole Fonksiyonları

Bir Boole fonksiyonu şunlara sahiptir:

- En az bir Boole değişkeni,
- En az bir Boole operatörü ve • {0,1}

kümesinden en az bir girdi.

{0,1} kümesinin bir üyesi olan bir çıktı üretir .

Artık ikili sayı sisteminin dijital sistemlerde neden bu kadar kullanışlı olduğunu biliyorsunuz .

Boole Cebiri – Boole Fonksiyonları

Gerçek tablosu

Boole fonksiyonu:

$$F(x, y, z) = x\bar{z}+y$$

sağda gösterilmektedir.

Boole fonksiyonunun değerlendirilmesini kolaylaştırmak için doğruluk tablosu, fonksiyonun alt parçalarının değerlendirmelerini tutmak üzere ek (gölgeli) sütunlar içerir.

$$F(x, y, z) = x\bar{z}+y$$

x	y	z	$\bar{z}$	$x\bar{z}$	$x\bar{z}+y$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

Boole Cebiri – Öncelik kuralları

Yaygın olarak kullanılan aritmetik, Boole işlemlerinin öncelik kuralları vardır

NOT operatörü en yüksek önceliğe sahiptir , bunu AND ve en son da OR takip eder.

Tablomuzdaki (gölgeli) fonksiyon alt parçalarını bu şekilde seçtik .

$$F(x,y,z) = x\bar{z}+y$$

x	y	z	$\bar{z}$	$x\bar{z}$	$x\bar{z}+y$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

Boole Cebiri – Basitleştirme

Dijital bilgisayarlar, aşağıdakileri uygulayan devreler içerir: Boole fonksiyonları.

Boolean fonksiyonunu ne kadar basit hale getirebilirsek, ortaya çıkacak devre ne kadar küçük olursa.

Daha basit devreler, karmaşık devrelere göre daha ucuza inşa edilir, daha az güç tüketir ve daha hızlı çalışır.

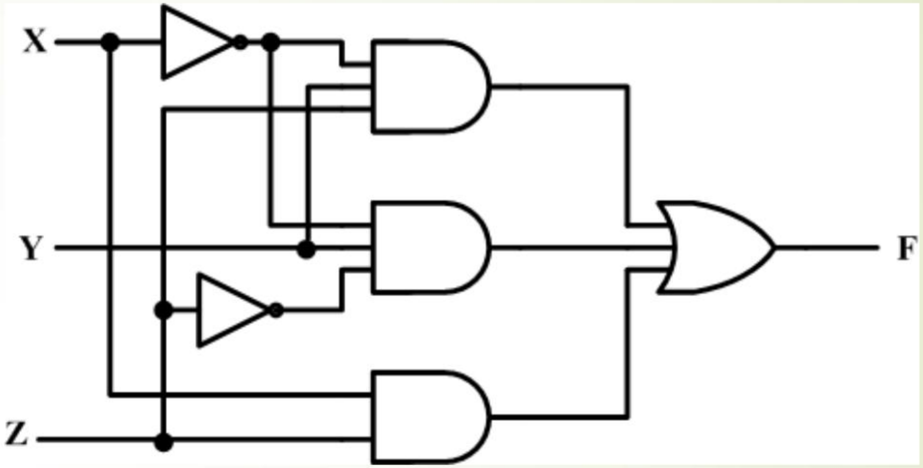
Bunu aklımızda tutarak, Boole fonksiyonlarımızı her zaman en basit biçimine indirgemek isteriz .

Bize yardımcı olan bir dizi Boole kimliği vardır

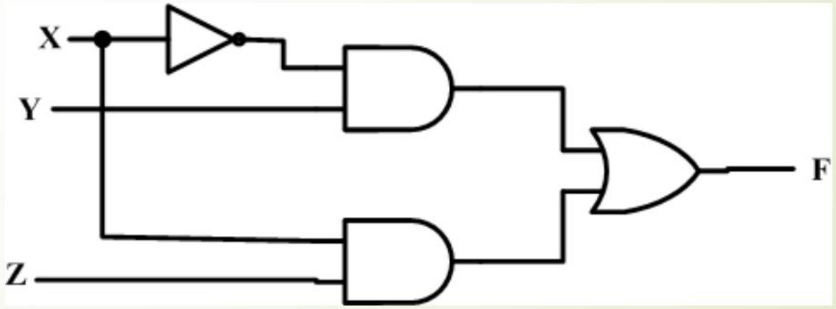
Bunu yapmak için.

Boole Cebiri – Uygulama üzerindeki etkisi

F = X'YZ + X'YZ' + XZ



F = X'Y + XZ'ye indirgenir



Boole Cebiri – Yasalar

Çoğu Boolean kimliği bir AND (ürün) formu ve bir OR (toplam) formuna sahiptir. Kimliklerimizi her iki formu kullanarak veririz. İlk grubumuz oldukça sezgiseldir:

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0 + x = x$
Null Law	$0x = 0$	$1 + x = 1$
Idempotent Law	$xx = x$	$x + x = x$
Inverse Law	$x\overline{x} = 0$	$x + \overline{x} = 1$

Boole Cebiri – Yasalar (Devamı)

İkinci grup Boole özdeşliklerimiz cebir çalışmalarınızdan size tanıdık gelecektir:

Identity Name	AND Form	OR Form
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x + (y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$

Dağıtım yasasını kullanarak $x+x'y=(x+x')(x+y)$

Boole Cebiri – Yasalar (Devamı)

Boole kimliklerinin son grubu belki de en faydalı olanlardır.

Eğer küme teorisini veya biçimsel mantığı incelediyseniz, bunlar Kanunlar da size tanıdık geliyor.

Identity Name	AND Form	OR Form
Absorption Law	$x(x+y) = x$	$x + xy = x$
DeMorgan's Law	$\overline{(xy)} = \overline{x} + \overline{y}$	$\overline{(x+y)} = \overline{x}\,\overline{y}$
Double Complement Law	$\overline{(\overline{x})} = x$	

Boole Cebiri – DeMorgan Yasası

Bazen bir ev inşa etmek daha ekonomiktir

Bir fonksiyonun tamamlayıcısını kullanan (ve sonucunu tamamlayan) bir devre, fonksiyonu doğrudan uygulamaktan daha iyidir.

DeMorgan yasası , bir Boole fonksiyonunun tamamlayıcısını bulmanın kolay bir yolunu sağlar .

DeMorgan yasasını hatırlayalım :

$$\overline{(xy)} = \overline{x} + \overline{y} \quad \text{and} \quad \overline{(x+y)} = \overline{x}\,\overline{y}$$

Boole Cebiri – İspat: DeMorgan

Teorem 3.7: DeMorgan Yasası

Boole cebirindeki her bir eleman çifti için:

(+) =

( ) = +

Kanıt:

(+ +) = + + (+) ( )

= (+ + +) ( )

= 1 (+) 1 + ( )

= 1 1

= 1

Not: Kullanarak
Her işlemin
birleştiriciliği (+),
( )

Boole Cebri – DeMorgan Yasası (Devamı)

DeMorgan yasası herhangi sayıda değişkene genişletilebilir değişkenler.

Her değişkeni tamamlayıcısı ile değiştirin ve tüm VE'leri VEYA'ya, tüm VEYA'ları VE'ye dönüştürün.

Böylece, şunun tamamlayıcısını buluruz :

$$F(X,Y,Z) = (XY) + (\overline{X}Z) + (Y\overline{Z})$$

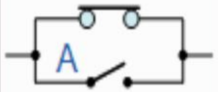
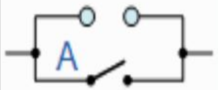
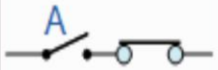
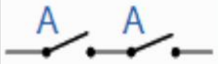
şudur:

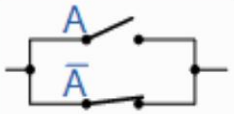
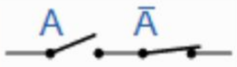
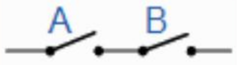
$$\overline{F}(X,Y,Z) = \overline{(XY) + (\overline{X}Z) + (Y\overline{Z})}$$

$$= \overline{(XY)}\,\overline{(\overline{X}Z)}\,\overline{(Y\overline{Z})}$$

$$= (\overline{X}+\overline{Y})(X+\overline{Z})(\overline{Y}+Z)$$

Boole Yasaları için Doğruluk Tabloları

Boolean İfade	Tanım	Eş değer Geçiş Devre	Boole Cebiri Kanun veya Kural
Bir + 1 = 1	A ile paralel olarak kapalı = "KAPALI"		İptal
Bir + 0 = Bir	A ile paralel olarak açık = "A"		Kimlik
A . 1 = Bir	Seri halinde bir kapalı = "A"		Kimlik
Bir . 0 = 0	Seri halinde bir açık = "AÇIK"		İptal
A + A = A	A ile paralel olarak Bir = "Bir"		İdempotent
A . Bir = Bir	Seri halinde bir Bir = "Bir"		İdempotent

Boolean İfade	Tanım	Eş değer Geçiş Devre	Boole Cebiri Kanun veya Kural
=	DEĞİL DEĞİL A (çift olumsuzluk) = "A"		Çift Olumsuzluk
+ =	A ile paralel olarak NOT A = "KAPALI"		Tamamlayıcı
. =	Seri halinde bir NOT A = "AÇIK"		Tamamlayıcı
A+B = B+A	A, B ile paralel = B, A ile paralel		Değişmeli
AB = BA	A, B ile seri halinde = B, A ile seri halinde		Değişmeli
$\overline{(A+B)}$ = .	ters çevirin ve VEYA'yı VE ile değiştirin		de Morgan Teoremi
$\overline{(.)}$ = +	ters çevir ve VE'yi VEYA ile değiştir		de Morgan Teoremi

Boole Cebiri – Tamamlayıcı Fonksiyonlar

EXAMPLE 2.2

Find the complement of the functions $F_1 = x'yz' + x'y'z$ and $F_2 = x(y'z' + yz)$. By applying DeMorgan's theorems as many times as necessary, the complements are obtained as follows:

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$$

$$\begin{aligned} F_2' = [x(y'z' + yz)]' &= x' + (y'z' + yz)' = x' + (y'z')'(yz)' \\ &= x' + (y + z)(y' + z') \\ &= x' + yz' + y'z \end{aligned}$$

Boole Cebiri – İkililik İlkesi

Bir fonksiyonun duali ne anlama gelir?

Bir fonksiyonun ikiliği, VEYA ve VE işlemlerinin yer değiştirilmesi ve 1'ler ve 0'ların 0'lar ve 1'ler ile değiştirilmesiyle elde edilir .

Fonksiyon tamamlayıcısını elde etmenin kısayolu

F=X'YZ'+X'Y'Z

F=(X'+Y+Z')(X'+Y'+Z) ikilisini üret Her bir

sabiti tamamlayarak şunu elde et:

F'=(X+Y'+Z)(X+Y+Z')

Her varsayım iki ifadeden oluşur; bir ifade

(+) ve ( ) işlemleri ile 0 ve 1 kimlik elemanlarının yer değiştirilmesiyle diğerine dönüştürülür.

Bu tür ifadeler birbirinin ikiliği olarak bilinir. Eğer bir eşdeğerlik

kanıtlanırsa, o zaman ikiliği de hemen doğru olur. Örneğin: + = 1'i kanıtlarsak ,

ikiliğe göre: + ( ) (¯ ¯) ( )

(¯ + = 0)

Boole Cebiri – Dualite İlkesi (Devamı)

Bir A ifadesini eşdeğer bir B ifadesine dönüştürmemize yardımcı olabilecek Boole ifadelerinin yararlı kimlikleri vardır

Boolean ifadesinin duali yardımıyla ek kimlikler türetebiliriz .

Bir Boole ifadesinin ikiliği, Boole toplamları ve Boole çarpımları ve 0'lar ve 1'ler yer değiştirilerek elde edilir.

22 Boole Cebiri – Dualite

Örnekler:

x(y + z)'nin ikilisi şudur: x + yz

x 1 + (y + z)'nin duali şudur: (x + 0)(yz)

Bir Boole ifadesi ile gösterilen bir Boole fonksiyonu F'nin duali , bu ifadenin duali ile gösterilen fonksiyondur .

Fd ile gösterilen bu ikili fonksiyon, F'yi temsil etmek için kullanılan belirli Boole ifadesine bağlı değildir.

Boole Cebiri – Dualite

Bu nedenle, Boolean ifadeleriyle temsil edilen fonksiyonlar arasındaki bir özdeşlik, özdeşliğin her iki tarafının dualleri alındığında geçerliliğini korur .

İkilik ilkesi adı verilen bu gerçeği, yeni kimlikler türetmek için kullanabiliriz .

Örneğin, emilim yasasını $x(x + y) = x$ olarak ele alalım

Bu özdeşliğin her iki tarafının dualleri alındığında, $x + xy = x$ denklemi elde edilir; bu da bir özdeşliktir (ve aynı zamanda bir emilim yasası olarak da adlandırılır).

Boole Cebiri – Örnek

Basitleştirmek için Boole kimliklerini kullanabiliriz

işlev: $F(X,Y,Z) = (X + Y)(X + \overline{Y})(\overline{X\overline{Z}})$

aşağıdaki gibi:

$(X + Y)(X + \overline{Y})(\overline{X\overline{Z}})$	Idempotent Law (Rewriting)
$(X + Y)(X + \overline{Y})(\overline{X} + Z)$	DeMorgan's Law
$(XX + X\overline{Y} + XY + Y\overline{Y})(\overline{X} + Z)$	Distributive Law
$((X + Y\overline{Y}) + X(Y + \overline{Y}))(\overline{X} + Z)$	Commutative & Distributive Laws
$((X + 0) + X(1))(\overline{X} + Z)$	Inverse Law
$X(\overline{X} + Z)$	Idempotent Law
$X\overline{X} + XZ$	Distributive Law
$0 + XZ$	Inverse Law
XZ	Idempotent Law

Boole Cebiri Basitleştirme Örneği

S = (A + B).(A + C)

AA + AC + AB + BC – Dağıtım yasası

A + AC + AB + BC – İdempotent VE yasası (AA = A)

A(1 + C) + AB + BC – Dağıtım yasası

A.1 + AB + BC – Kimlik VEYA yasası (1 + C = 1)

A(1 + B) + BC – Dağıtım yasası

A.1 + M.Ö. – Kimlik VEYA yasası (1 + B = 1)

S = A + (MK) – Kimlik VE yasa (A.1 = A)

Boole Cebiri – Özet

Table 2.1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

Boole Cebiri – Basitleştirme

Simplify the following Boolean functions to a minimum number of literals.

1. $x(x' + y) = xx' + xy = 0 + xy = xy.$

2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y.$

3. $(x + y)(x + y') = x + xy + xy' + yy' = x(1 + y + y') = x.$

4.
$$\begin{aligned} xy + x'z + yz &= xy + x'z + yz(x + x') \\ &= xy + x'z + xyz + x'yz \\ &= xy(1 + z) + x'z(1 + y) \\ &= xy + x'z. \end{aligned}$$

5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$, by duality from function 4.

Boole Cebiri – Basitleştirme

Aşağıdaki denklemlerin basitleştirilmiş eşdeğerini bulun

a(a+b')

a.a+a.b'=a+a.b'=a.(1+b)=a.1=a

a+(a'+b)'

a+(a')'.b'=a+ab'=a

x.y+x'.z+y.z+y'z

xy+z

xyz+x'yz+y'z

z

Boole Cebiri – Basitleştirme

A.(A'+B)=?

(A+B).(A+C)=?

F=A'.B+A+AB=?

A.B+A'C+BC

A.B+A'.C

A'B'C' + A'B'C + ABC' + AB'C'

Boole Cebiri – Basitleştirme

$F(A,B,C) = A'B'C' + A'B'C + AB'C' + AB'C + ABC' + ABC$

$=A'B'(C'+C)+AB'(C'+C)+AB(C'+C)$

$=A'B'+AB'+AB$

$= A'B'+AB'+AB+AB'$

$=B'(A'+A)+A(B+B')$

$=B'+A$

Boole Cebiri – Basitleştirme

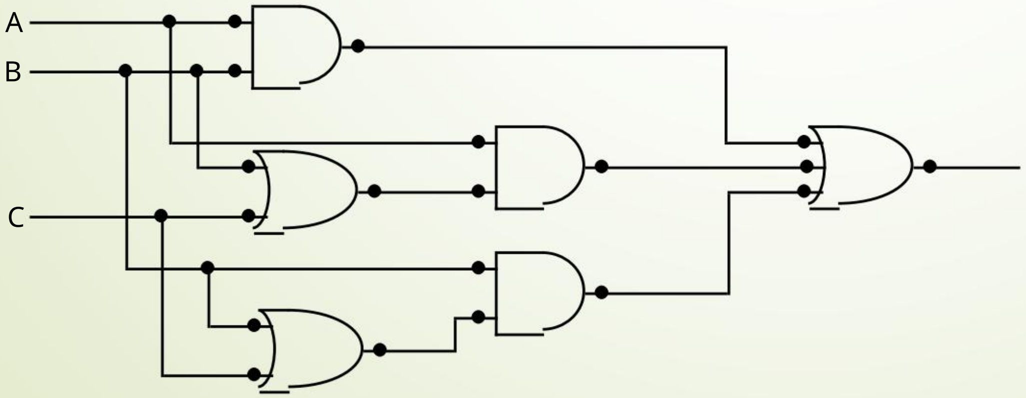
$AB + A(B + C) + B(B + C)$

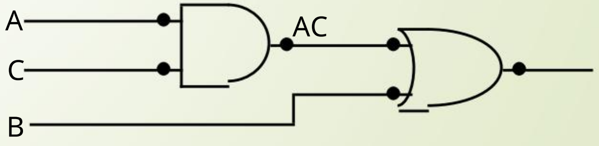
$= AB + AB + AC + BB + BC$

$= AB + AC + B + BC$

$= AB + AC + B$

$=AC + B$





Boole Cebiri – Mantıksal Olarak Eşdeğer İfadeler

Boole ifadelerini basitleştirme alıştırmalarımızda, aynı Boole ifadesini ifade etmenin çok sayıda yolu olduğunu görüyoruz.

Bu "eş anlamlı" biçimler mantıksal olarak eşdeğerdir.

Mantıksal olarak eşdeğer ifadeler aynı doğruluğa sahiptir tablolar.

Mümkün olduğunca fazla karışıklığı ortadan kaldırmak için, Tasarımcılar Boole fonksiyonlarını standartlaştırılmış veya kanonik biçimde ifade ederler.

Boole Cebiri – Ürünlerin Toplamı ve Toplamların Ürünü

Boole ifadeleri için iki kanonik form vardır : toplam-ürünlerin-ürünleri ve toplamların-ürünü.

Boole ürününün AND işlemi olduğunu ve

Boolean toplamı VEYA işlemidir.

Ürünlerin toplamı biçiminde , AND'li değişkenler OR'lidir birlikte.

Örneğin: $F(x,y,z) = xy + xz + yz$

Toplamların çarpımı biçiminde , OR'lu değişkenler AND'lidir birlikte:

Örneğin: $F(x,y,z) = (x+y)(x+z)(y+z)$

Boole Cebiri – Normal Formlar

Şu fonksiyonu ele alalım:

$$(\quad , \quad , \quad , \quad) = + +$$

Bir literal , bir formülde tamamlayıcı veya tamamlayıcı olmayan bir değişkenin bulunmasıdır.

Bir ürün terimi, bir kelimenin tam anlamıyla veya bir ürünü (bağlacı) olabilir.

harfi harfine.

Ayrık normal form: Bir Boole formülü şu şekilde yazılır: tek ürün terimi veya ürün terimlerinin toplamı (ayrılması) olarak.

Boole Cebiri – Normal Formlar

Şu fonksiyonu göz önünde bulundurun:

$$(\quad , \quad , \quad , \quad) = (\; + \;)(\; + \; + \;)$$

Bir toplam terimi, bir tam sayının veya bir toplamın (ayrılmanın) terimidir.

harfi harfine.

Bağlaçlı normal form: Tek bir toplam terimi veya toplam terimlerinin bir ürünü (bağlaç) olarak yazılan bir Boole formülü.

Boole Cebiri – Kanonik Formüller

Doğruluk tablosu verildiğinde Boole formülü nasıl elde edilir?

X	Evet	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Boole Cebiri – Minterm Kanonik Formül

X	Evet	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

(, ,) = + +

Boole Cebiri – m-Gösterimi

X	Evet	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

(, ,) şöyle yazılabilir (, ,) = $_1$ + $_3$ + $_4$

(, ,) = Σ (1,3,4)

Boole Cebiri – Maxterm Kanonik Formül

X	Evet	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

+ +

+ +

+ +

+ +

+ +

(, ,) = +(+ (+ +)(+)(+ +)

+)(+ +)

Boole Cebiri – Ürünlerin Toplamı

Bir fonksiyonun doğruluk tablosunu kullanarak onu ürünlerin toplamı biçimine dönüştürmek kolaydır .

Fonksiyonu doğru yapan değişkenlerin değerleriyle ilgileniyoruz (=1).

Gerçeklik tablosunu kullanarak, şunları listeliyoruz:
Gerçek bir fonksiyon değeriyle sonuçlanan değişkenlerin değerleri.

Daha sonra değişkenlerin her grubu birlikte VEYAlanır.

$$F(x,y,z) = x\bar{z}+y$$

x	y	z	$x\bar{z}+y$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Boole Cebiri – Ürünlerin Toplamı

Fonksiyonumuz için ürünlerin toplamı formu şu şekildedir:

$$F(x,y,z) = \bar{x}y\bar{z}+\bar{x}yz+x\bar{y}\bar{z}+xy\bar{z}+xyz$$

Bu fonksiyonun en basit terimlerle olmadığını fark ediyoruz . Amacımız sadece fonksiyonumuzu kanonik ürün toplamı biçiminde yeniden yazmaktır.

$$F(x,y,z) = x\bar{z}+y$$

x	y	z	$x\bar{z}+y$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

CNF ve DNF

Aşağıdaki doğruluk tabloları için hem DNF hem de CNF'yi kullanarak f fonksiyonunu bulun

X	Y	*f*
0	0	0
0	1	1
1	0	1
1	1	0

X	Y	*f*
0	0	1
0	1	1
1	0	1
1	1	1

X	Y	Z	*f*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Mantık Kapıları

Boolean fonksiyonlarını soyut terimlerle inceledik.

Bu bölümde, Boole fonksiyonlarının

Kapı adı verilen dijital bilgisayar devrelerinde uygulanır .

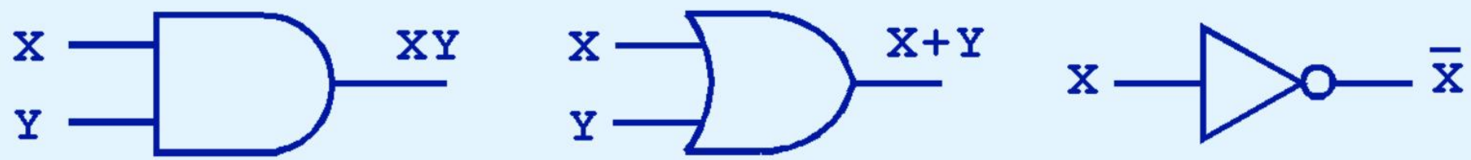
Kapı, iki veya daha fazla giriş değerine dayalı olarak bir sonuç üreten elektronik bir cihazdır.

Gerçekte kapılar bir ila altı transistörden oluşur, ancak dijital tasarımcılar bunları tek bir birim olarak düşünürler.

Entegre devreler, aşağıdakilere uygun kapı koleksiyonları içerir: belirli bir amaç.

Mantık Kapıları – VE VEYA DEĞİL

En basit üç kapı AND, OR ve NOT'tur.
kapılar.



X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

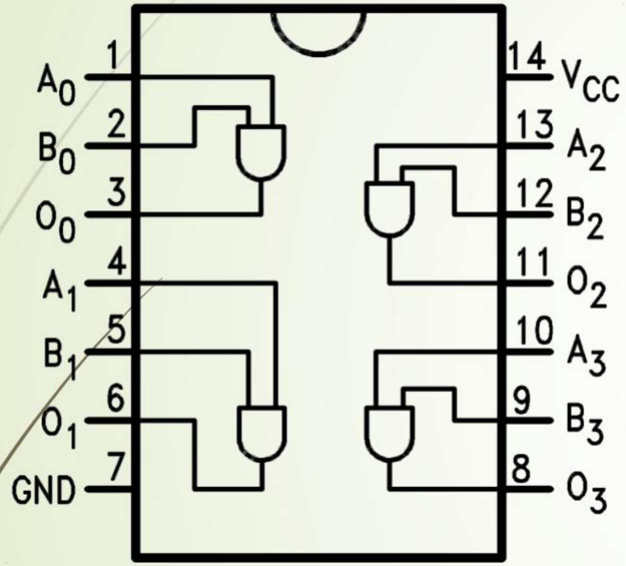
X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT X

X	$\overline{X}$
0	1
1	0

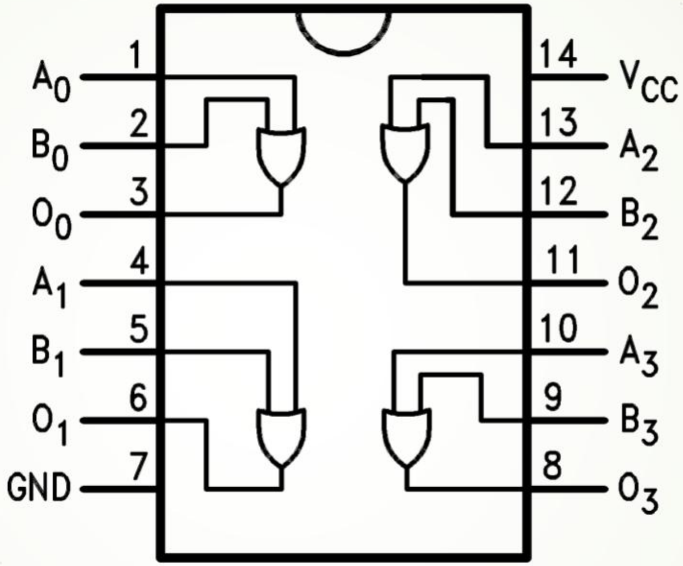
Bunlar, doğruluk tablolarından da görebileceğiniz gibi, doğrudan ilgili Boole işlemlerine karşılık gelir.

Mantık Kapıları – VE VEYA DEĞİL



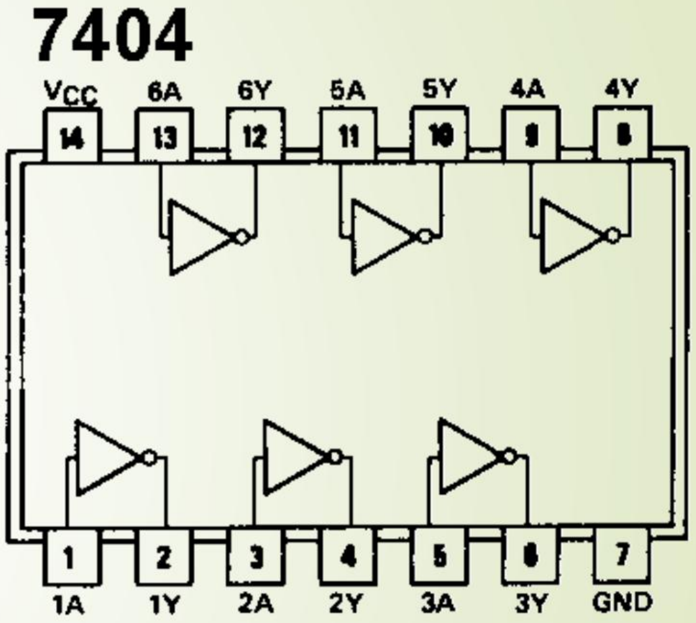
74AC08, 74ACT08

Quad 2-Girişli VE Kapısı



74AC32, 74ACT32

Dörtlü 2-Girişli VEYA Kapısı



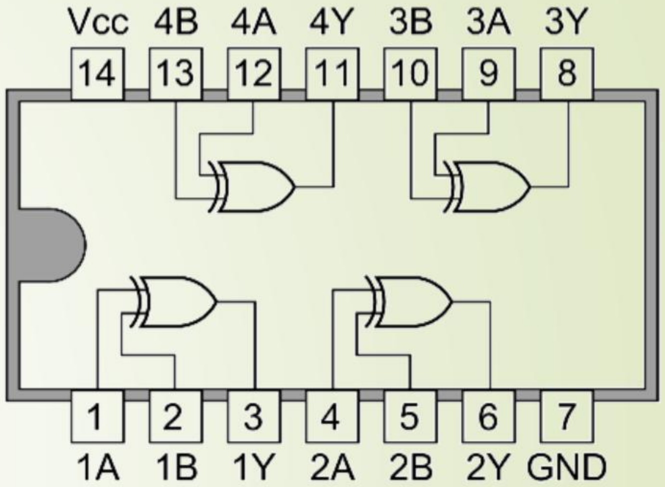
7404

6-Girişli DEĞİL Kapısı

Mantık Kapıları – XOR

Bir diğer çok kullanışlı kapı ise özel VEYA (XOR) kapısıdır.

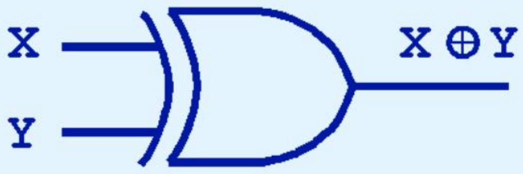
XOR işleminin çıktısı yalnızca giriş değerleri farklı olduğunda doğrudur.



74HCT86 Dörtlü 2 giriş ÖZEL-VEYA kapısı

X XOR Y

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

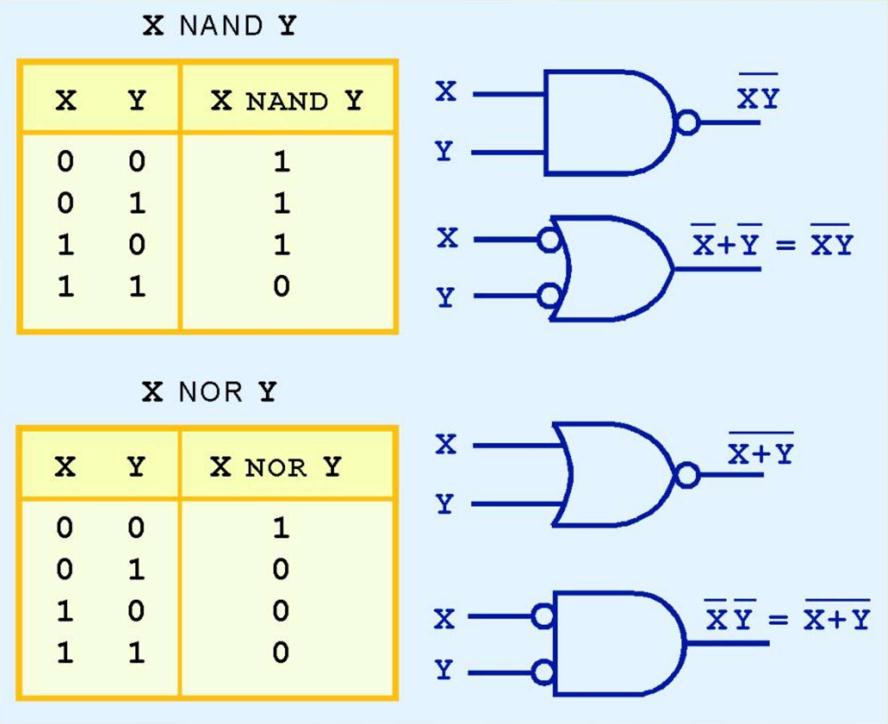


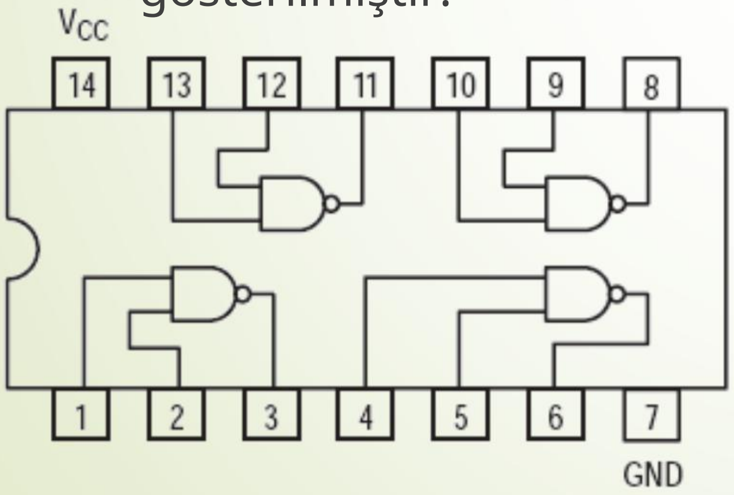
XOR işlemi için özel sembol 'ye dikkat edin .

Mantık Kapıları – NAND NOR

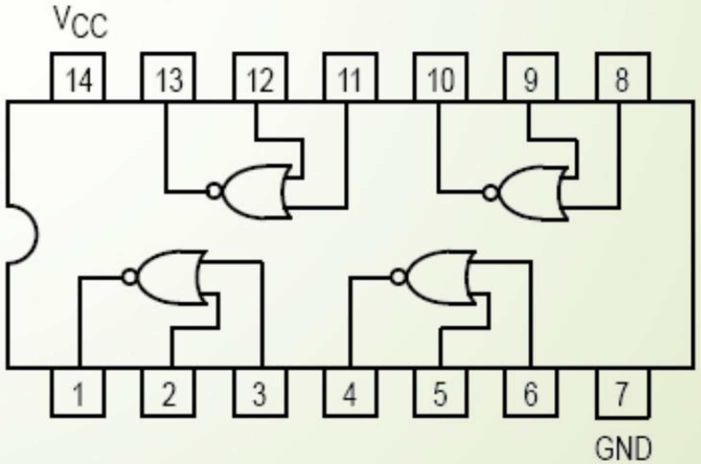
NAND ve NOR iki çok önemli kapıdır.

Sembolleri ve doğruluk tabloları sağda gösterilmiştir.





7400 - Dörtlü 2 Giriş NAND Kapısı



74LS02 - Dörtlü 2 Girişli NOR Geçit

Mantık Kapıları

NAND ve NOR

evrensel kapılar

olarak bilinir

çünkü üretimi ucuzdur

ve herhangi bir

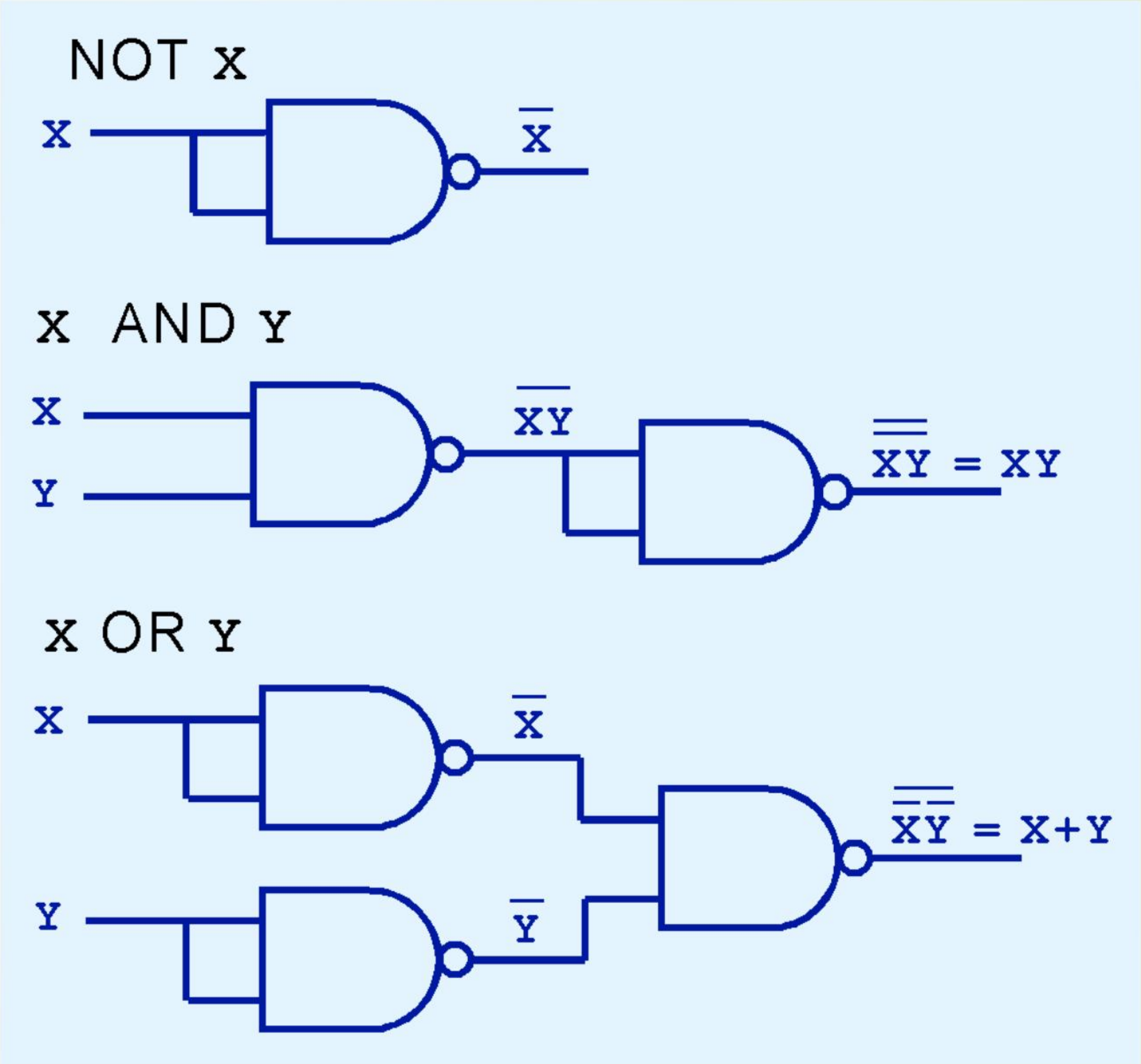
Boole fonksiyonu

yalnızca NAND

veya yalnızca NOR

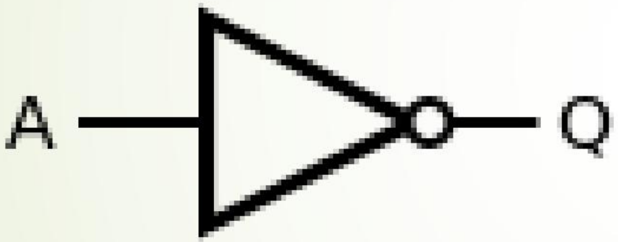
kapıları kullanılarak

oluşturulabilir .

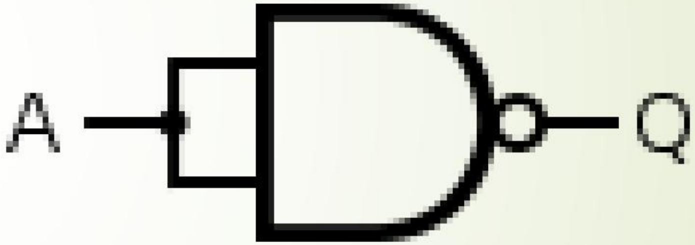


Mantık Kapıları NAND ile NOT kapısı

Hedef



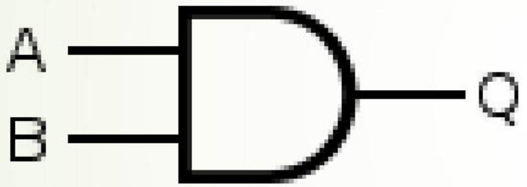
NAND kullanımı



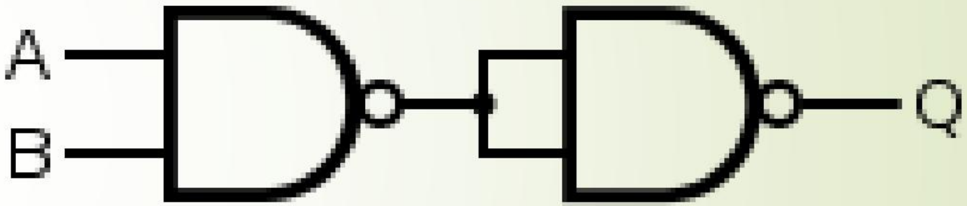
Gerçeklik Tablosu	
Giriş A	Çıkış Q
0	1
1	0

Mantık Kapıları NAND ile AND

Hedef

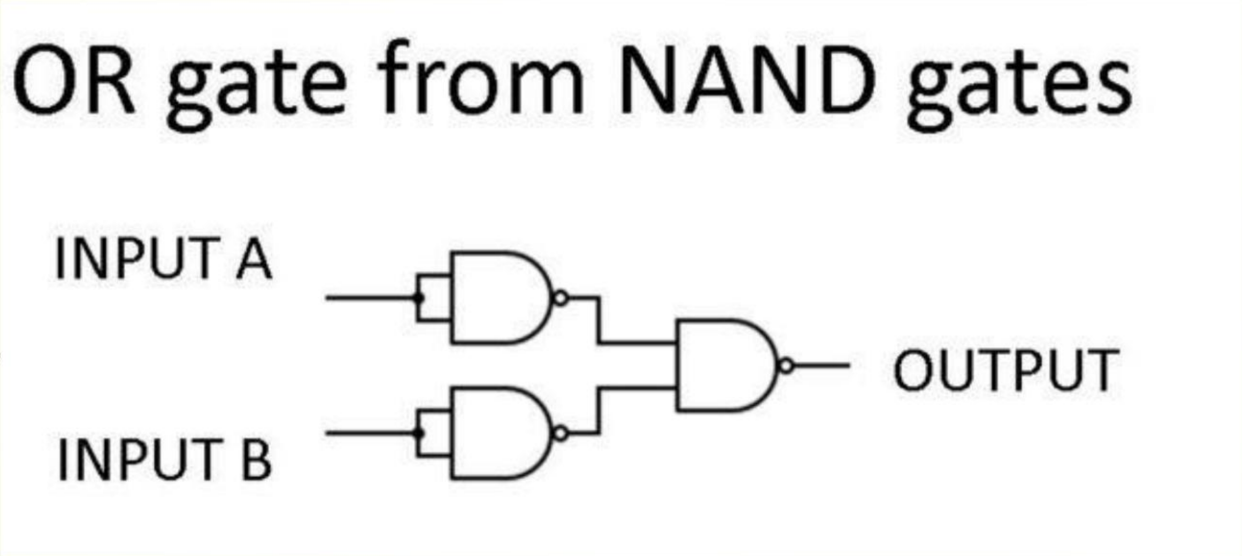


NAND kullanımı



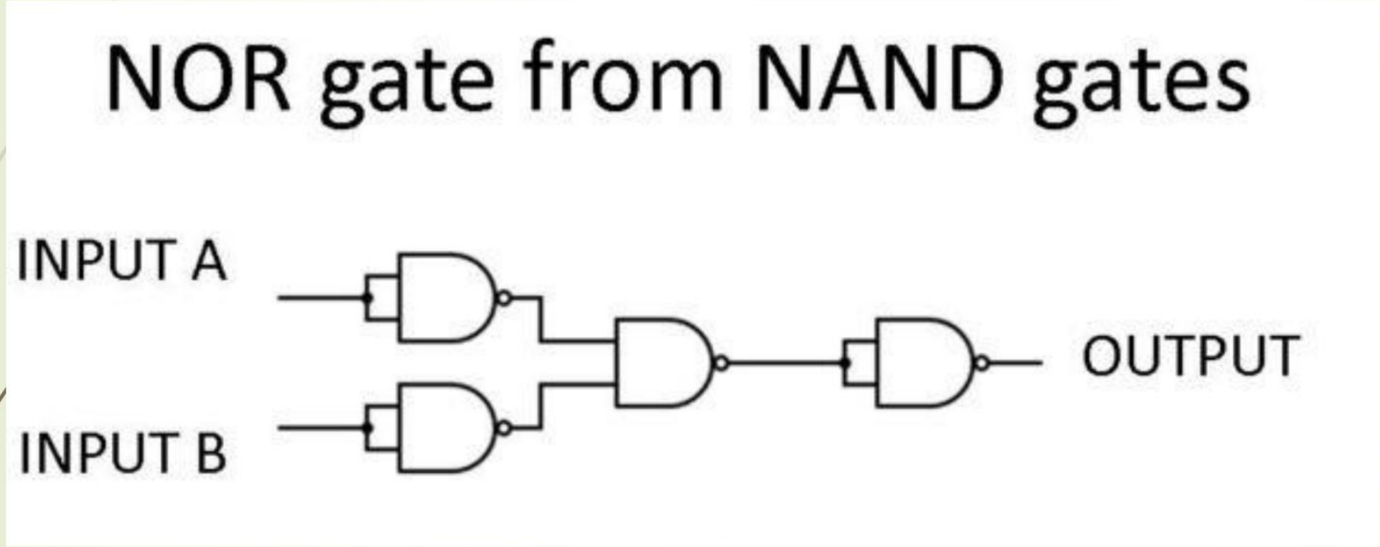
Gerçeklik Tablosu		
Giriş A	Giriş B	Çıkış Q
0	0	0
0	1	0
1	0	0
1	1	1

Mantık Kapıları NAND ile VEYA



NAND (VE değil)		
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

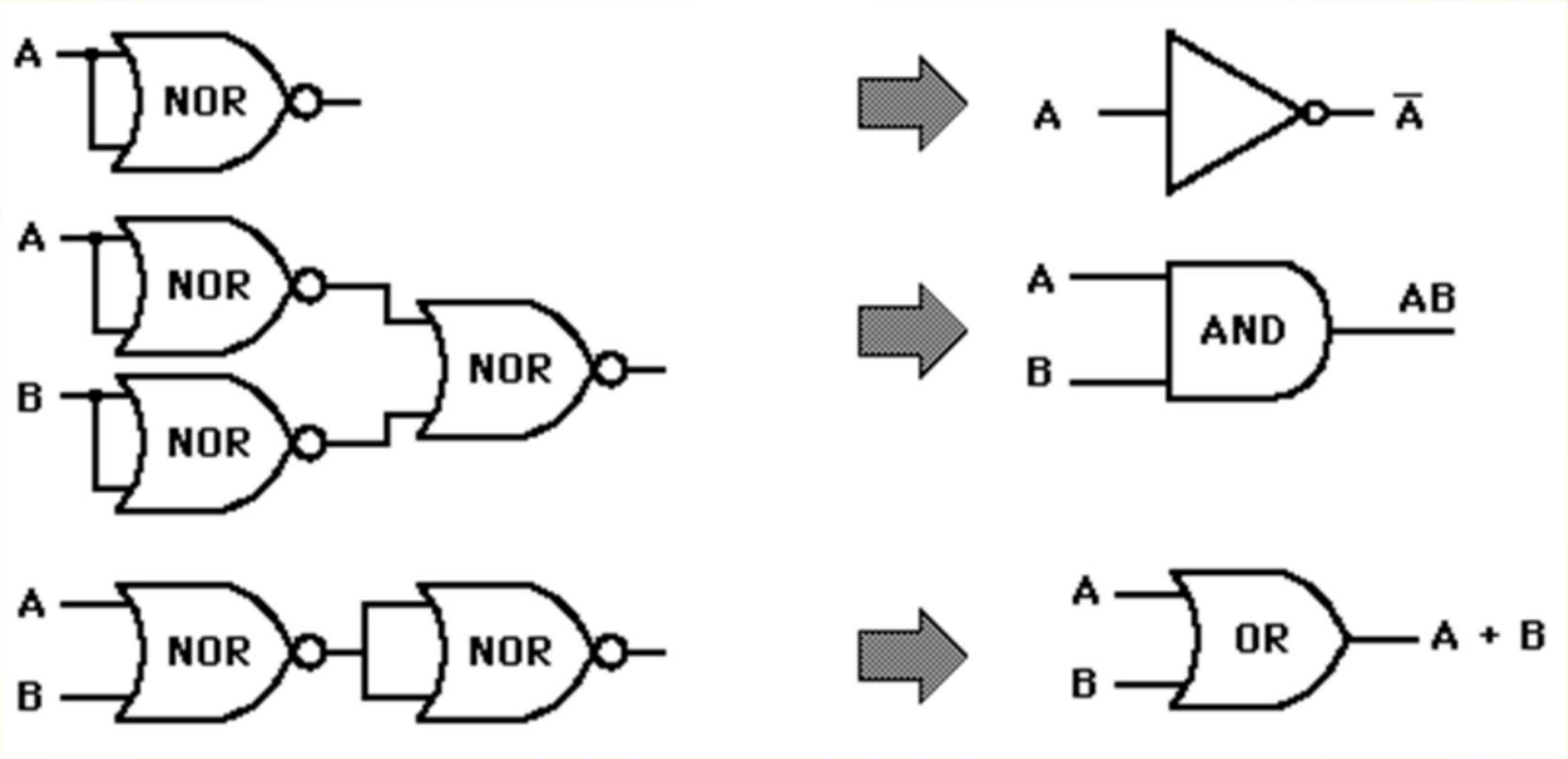
Mantık Kapıları NAND ile NOR Kapısı



NAND (VE değil)		
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

NE DE		
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Mantık Kapıları – NOR kapısını kullanarak diğer kapıları oluşturma

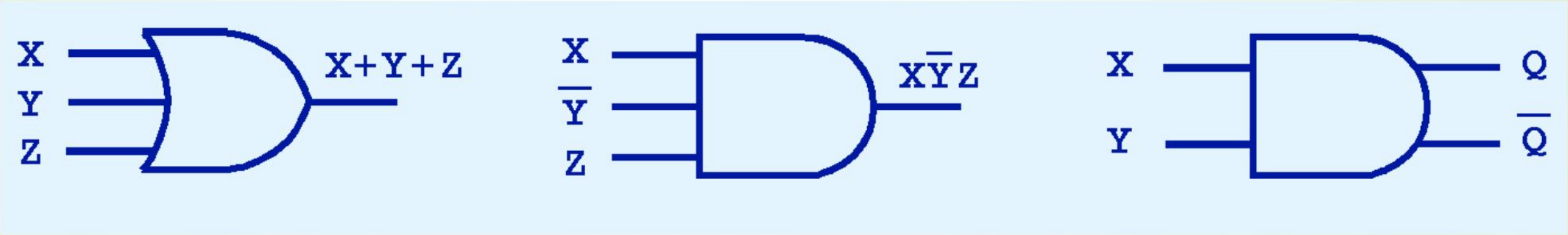


Mantık Kapıları

Kapılar birden fazla girişe ve birden fazla çıkışa sahip olabilir .

İşlemin tamamlayıcısı olarak ikinci bir çıktı sağlanabilir .

Bunun daha fazlasını ileride göreceğiz.



Mantık Kapıları – Örnek 1

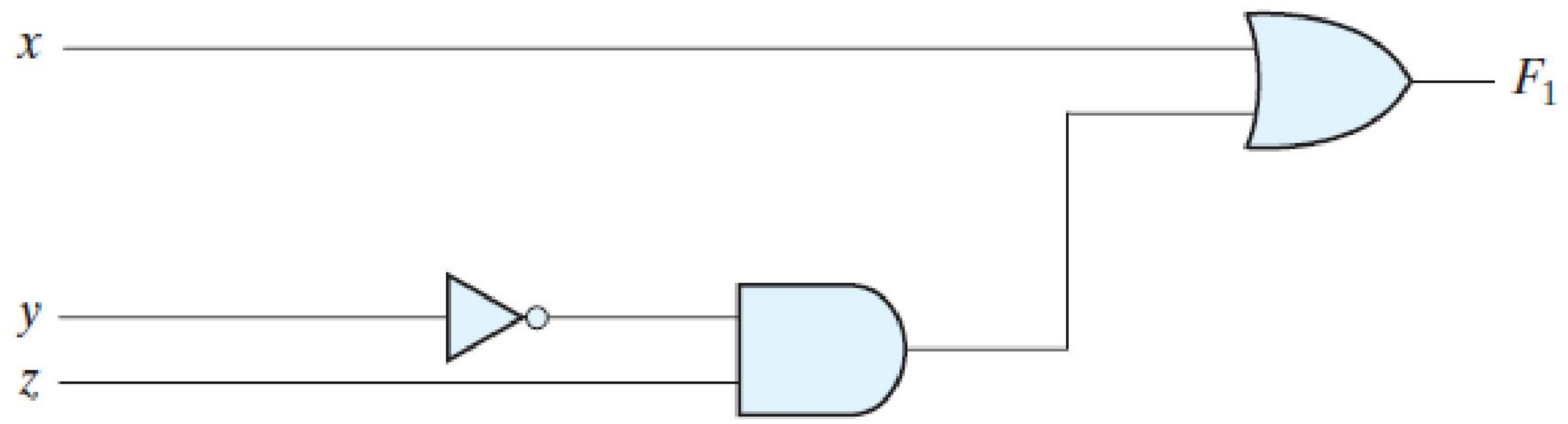
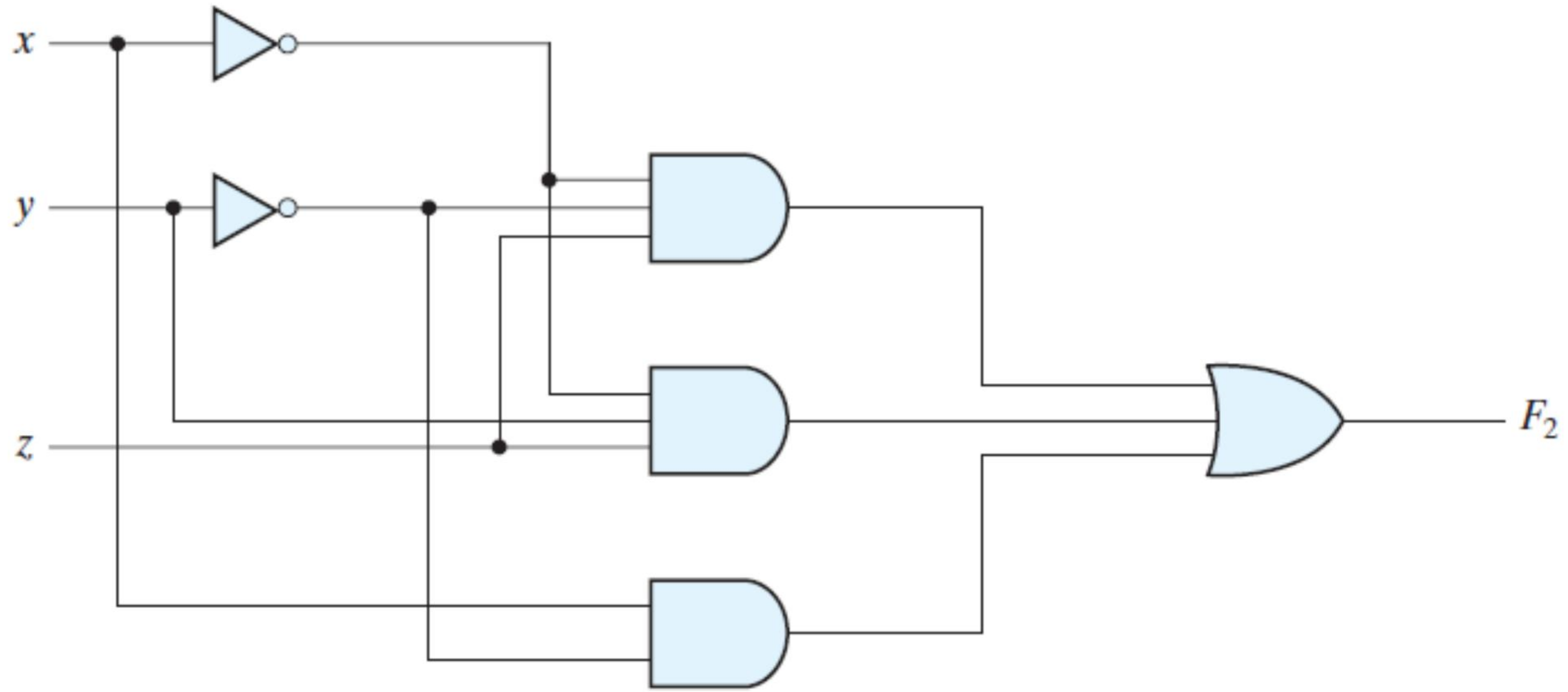


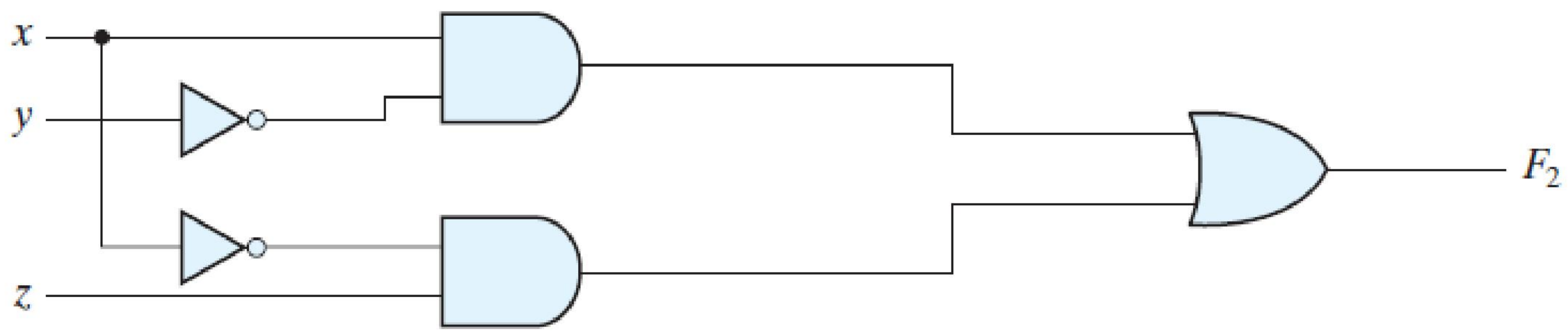
FIGURE 2.1
Gate implementation of $F_1 = x + y'z$

Mantık Kapıları – Örnek 2



(a) $F_2 = x'y'z + x'yz + xy'$

Mantık Kapıları – Örnek 3



(b) $F_2 = xy' + x'z$